Statistical Learning Theory (1)

Manel Martínez-Ramón

ECE, UNM

October, 2018

The MMSE criterion

The Minimum mean square criterion leads to the delta rule algorithm

- The idea of the Delta rule is quite simple.
- Assume a set of data represented in (column vectors) $\mathbf{x}_i$ of dimension D and a linear function

$$f(\mathbf{x}_n) = \hat{y}_n = \mathbf{w}^\top \mathbf{x}_n + b$$

where $\mathbf{w}$ is a column vector containing D parameters w_j, this is

$$\mathbf{w} = \{w_1 \cdots w_D\}^\top$$

and therefore

$$\mathbf{x}_n = \{x_{1,n} \cdots x_{D,n}\}^\top$$

- In order to get rid of the bias, we modify the notation as

$$\mathbf{w} = \{w_1 \cdots w_D \; w_{D+1}\}^\top$$

$$\mathbf{x}_n = \{x_{1,n} \cdots x_{D,n} \; 1\}^\top$$

Then

$$\hat{y}_n = \mathbf{w}^\top \mathbf{x}_n = \sum_{j=1}^{D} w_n x_{j,n} + w_{D+1}$$

Here w_{D+1} plays the role of b

- Assume now a set of labels for the data, this is, we construct a set

$$\mathcal{D} : \{\mathbf{x}_n,\ y_n\}, 1 \leq n \leq N$$

Criterion: to minimize the mean squared error of the estimation:

$$\mathcal{L}(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \mathbb{E}\left[\left(y_n - \mathbf{w}^\top \mathbf{x}_n\right)^2\right]$$

with respect to $\mathbf{w}$.
$\mathbf{X}$ and $\mathbf{y}$ are a matrix and a vector containing the data and the labels.

- This can be approximated as

$$\begin{aligned}\mathcal{L}(\mathbf{X},\mathbf{y},\mathbf{w}) &= \frac{1}{N}\sum_{i=1}\left(y_n-\mathbf{w}^\top\mathbf{x}_n\right)^2\\ &= \frac{1}{N}\sum_{i=1}\left(y_n^2+\mathbf{w}^\top\mathbf{x}_n\mathbf{x}_n^\top\mathbf{w}-2y_n\mathbf{w}^\top\mathbf{x}_n\right)\\ &= \frac{1}{N}\left(\sum_{i=1}y_n^2+\mathbf{w}^\top\mathbf{X}\mathbf{X}^\top\mathbf{w}-2\mathbf{w}^\top\mathbf{X}\mathbf{y}\right)\\ &= \frac{1}{N}\sum_{i=1}y_n^2+\mathbf{w}^\top\mathbf{R}\mathbf{w}-2\mathbf{w}^\top\mathbf{p}\end{aligned}$$

where $\mathbf{R}$ and $\mathbf{p}$ are the estimates of the data autocorrelation matrix and the cross correlation vector between data and labels.

- The optimization is performed by computing the gradient wrt $\mathbf{w}$:

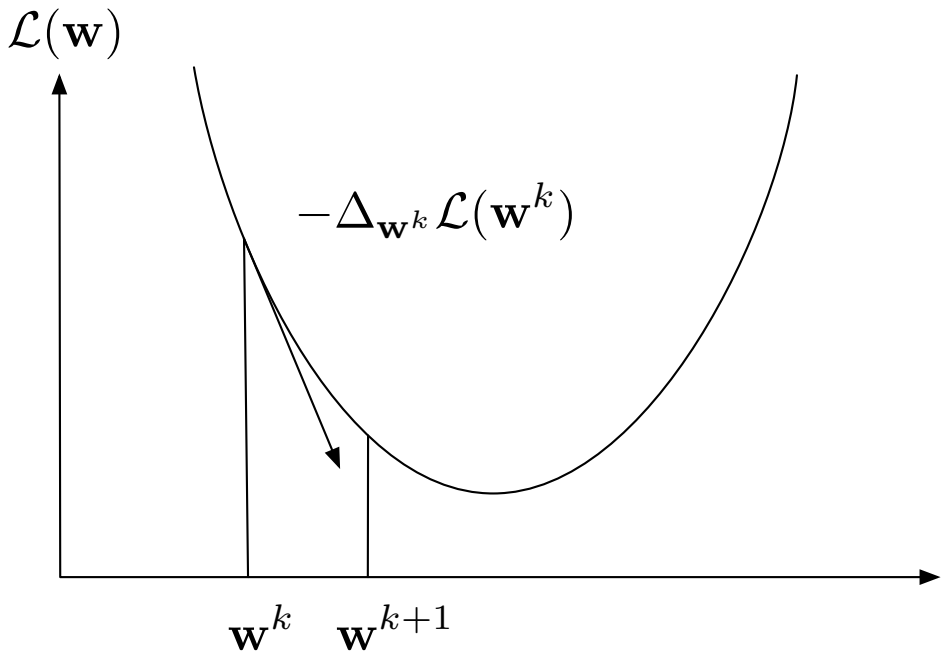
$$\Delta_{\mathbf{w}}\mathcal{L}(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \mathbf{R}\mathbf{w} - \mathbf{p}$$

If we null this gradient, we get the set of Widrow-Hoff equations:

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p}$$

Alternatively, we may want to proceed with a steepest descent procedure, where $\mathbf{w}$ is iteratively optimized by changing it in the direction opposite to its gradient. This gives rise to the Least Mean Squares algorithm.

The idea consists on updating the parameters towards the maximum descent of the gradient



$$\mathbf{w}^{k+1} = \mathbf{w}^k - \mu \Big[\mathbf{R}\mathbf{w}^k - \mathbf{p}\Big]$$

- The optimization rule is then

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \mu \left[\mathbf{R}\mathbf{w}^k - \mathbf{p} \right]$$

- But it may be the case that the samples come one at a time or that computing $\mathbf{R}$ and $\mathbf{p}$ is too complex or not convenient.
- Then we can approximate they by

$$\mathbf{R} \approx \mathbf{x}_n \mathbf{x}_n^\top$$

$$\mathbf{p} \approx \mathbf{x}_n y_n$$

- In that case the rule is

$$\begin{aligned}\mathbf{w}^{k+1} &= \mathbf{w}^k - \mu\left[\mathbf{x}_n\mathbf{x}_n^\top\mathbf{w}^k - \mathbf{x}_n y_n\right] \\ &= \mathbf{w}^k - \mu\left[\mathbf{x}_n^\top\mathbf{w}^k - y_n\right]\mathbf{x}_n \\ &= \mathbf{w}^k - \mu e_n\mathbf{x}_n\end{aligned}$$

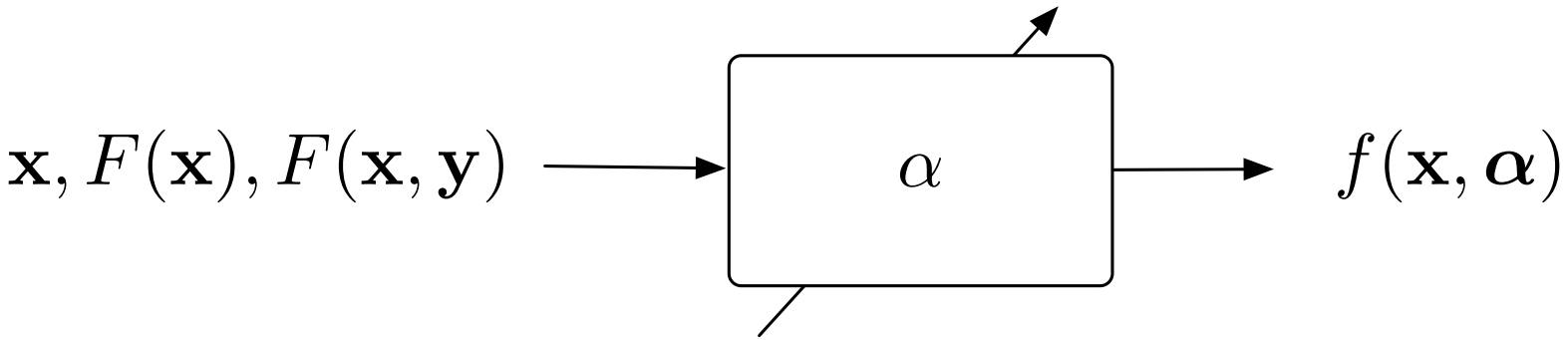
  where $e_n = (\mathbf{w}^k)^\top\mathbf{x}_n - y_n$ is the estimation error for data n at iteration k, and μ a small parameter.

- This is the Least Mean Squares Algorithm, the least computational burden algorithm to update the perceptron.

Risk minimization criteria

- We will assume that the data is distributed accordingly to a given probability distribution $F(\mathbf{x})$ which is unknown.
- Also, we assume that scatter values or labels y_n exist for each data $\mathbf{x}_n$, though they may be known or unknown.
- Therefore, a conditional distribution $F(\mathbf{x}|y)$ exists, and it is unknown.

- A learning machine is, from a purely abstract point of view, a machine capable to construct a set of parametric estimation functions $f(\mathbf{x}, \boldsymbol{\alpha})$ where $\boldsymbol{\alpha}$ is a set of parameters to be adjusted.



- In order to adjust the parameters, we can choose to minimize a given *risk function* with respect to them.
- Assume an estimation function $f(\mathbf{x}, \boldsymbol{\alpha})$ that estimates y, this is

$$f(\mathbf{x}_n, \boldsymbol{\alpha}) = y_n + e_n$$

  where e_n is the estimation error.
- Risk minimization criteria are those intended to minimize a convex function of the error. The risk function can be defined as

$$R(\boldsymbol{\alpha}) = \int_{\mathbf{x},y} L\left(y, f(\mathbf{x}, \boldsymbol{\alpha})\right) dF(\mathbf{x}, y)$$

Risk minimization criteria

- The risk function must have a minimum in the point where a function of the error is minimized over all samples.
- The loss function $L(\cdot)$ is a measure of discrepancy between the output and the label.
- Depending on the problem to solve, we want to choose one or another loss function.
- Learning is to minimize an estimate of the risk function when the distribution is unknown and the only available information is in the samples.

Empirical risk

- Since the previously seen risk cannot be computed, an approximation must be taken:

$$R_{emp}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} L\left(y, f(\mathbf{x}, \boldsymbol{\alpha})\right)$$

- This is called the **empirical** risk.
- The minimization of this risk can lead to *over-fitting* if the estimation function is complex enough.
- Thus, the complexity of the machine must be limited.
- The minimization of the complexity can be achieved using some optimizable measure of the complexity of the machine.

This is the idea of the Structural Risk Minimization (SRM).

Conclusion

In this lesson we have seen

- The MMSE optimization criterion.
- An algorithm that uses the MMSE criterion: the delta rule or Least Mean Squares.
- A more general definition of the risk minimization criteria.